100717-607/Bayer 10268 - CH

APPLICATION FOR

UNITED STATES LETTERS PATENT

# METHOD FOR DETERMINING THE PERMITTED WORKING RANGE OF A NEURAL NETWORK

Inventors:

MOGK, Georg
MRZIGLOD, Thomas
HÜBL, Peter

# Method for determining the permitted working range of a neural network

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

[0001] The invention relates to a method for checking whether an input data record is in the permitted working range of a neural network, and to a corresponding computer program product and system.

### 2. Description of the Related Art

[0002] A plurality of application possibilities for neural networks are known from the prior art. Neural networks are used for data-driven model formation, for example for physical, biological, chemical and technical processes and systems, cf. Babel W.: "Einsatzmöglichkeiten neuronaler Netze in der Industrie: Mustererkennung anhand überwachter Lernverfahren – mit Beispielen aus der Verkehrs- und Medizintechnik"[translation of title: "Possibilities of use of neural networks in industry: pattern recognition using monitored learning methods – with examples from transportation and medicine technology"], Expert Verlag, Renningen-Malmsheim, 1997. In particular, the fields of use of neural networks include process optimization, image processing, pattern recognition, robot control and medicine technology.

[0003] Before a neural network can be used for predictive or optimization purposes, it must be trained. This usually involves adapting the weights of the neurons by means of an iterative method using training data, cf. Bärmann F.: "Prozessmodellierung: Modellierung von Kontianlagen mit neuronalen Netzen" [translation of title: "Process modelling: modelling of continuous systems with neural networks"], Internet page NN-Tool, www.baermann.de and Bärmann F.: "Neuronale Netze". Skriptum zur Vorlesung, FH-Gelsenkirchen, Fachbereich Physikalische Technik, Fachgebiet Neuroinformatik [translation of title: "Neural networks". Lecture paper. Technical University of Gelsenkirchen, Department of Physico-Technology, Subject area of neurocomputing], 1998.

[0004] The so-called back propagation method is particularly suitable for training a neural network. A further approach is implemented in the program "NN-Tool 2000". This program is commercially available from Professor Frank Bärmann, Technical University of Gelsenkirchen, Department of Physical Technology. The corresponding training method is also described in the publication "Neural Network", Volume 5, pages 139 to 144, 1992, "On a class of efficient learning algorithms for neural networks", Frank Bärmann, Friedrich Biegler-König.

[0005] DE 195 31 967 discloses a method for training a neural network with the non-deterministic behaviour of a technical system. The neural network is integrated here into a control loop in such a way that the neural network outputs, as output variable, a manipulated variable to the technical system, and the technical system generates a controlled variable from the manipulated variable supplied by the neural network and said controlled variable is fed to the neural network as an input variable. Noise with a known noise distribution is superimposed on the manipulated variable before it is fed to the technical system. Further methods for training neural networks are known from DE 692 28 412 T2 and DE 198 38 654 C1.

[0006] In addition, a method for estimating the confidence level of the prediction which is output by a neural network is known from the prior art: Protzel P., Kindermann L., Tagscherer M., Lewandowski A. "Abschätzung der Vertrauenswürdigkeit von Neuronalen Netzprognosen bei der Prozessoptimierung [translated title: "Estimating the confidence level of neural network predictions when optimizing processes"], VDI Berichte [Reports] No. 1526, 2000. EP 0 762 245 B1 also discloses a method for recognizing faulty predictions in a neural-model-supported or neural process control system.

[0007] A common disadvantage of these methods known from the prior art is that they can only permit conclusions to be drawn about the sensitivity, in terms of variations of the training data, of the model which is made available by the neural network. However, it is thus not possible to draw conclusions about the confidence level of a prediction which is made by the neural network.

[0008] F. Bärmann; Handbuch zu NN-Tool 98 [translated title: "Neural Network Tool Manual"], 1998, discloses an approach in which it is attempted to estimate the prediction error at a specific point using the known prediction error at adjacent data points.

[0009] All these methods have in common the fact that it is not possible to draw a conclusion as to whether an input data record is at all in the permitted working range of the neural network. However, an incorrect estimation is possible only in this case.

[0010] The invention is therefore based on the object of providing a method which makes it possible to check whether an input data record is in the permitted working range of a neural network. In addition, the invention is based on the object of providing a corresponding computer program product.

[0011] The object on which the invention is based is achieved in each case with the features of the independent patent claims. Preferred embodiments of the invention are given in the dependent patent claims.

## SUMMARY OF THE INVENTION

[0012] The present invention makes it possible to check an input data record for a neural network to determine whether it is in the permitted working range of the neural network. The invention is based on the realization that structural information is not input into neural networks but instead only training input data records are used which have been, for example, obtained by measuring means. Due to this recognition, such models can provide trustworthy predictions only in the areas in which the models have been trained.

[0013] Between the given training data points it is possible to interpolate very efficiently using such models. However, in contrast to corresponding rigorous models, data-driven models cannot extrapolate, or can extrapolate only to a very restricted degree. Therefore, in particular for monitoring and/or controlling critical applications, it is advantageous that it is possible to check whether the model used is utilized in the permitted working range.

[0014] This applies to a greater degree also to hybrid models in which a plurality of neural networks is connected to rigorous models. Although hybrid models are capable of extrapolation as an overall model, the interpolation area must be checked for each individual data-driven subcomponent, that is to say, for the neural networks which are contained in the hybrid model.

[0015] According to the invention, the working range of a neural network is defined by the convex envelope formed by the training input data records of the neural network. For example, a neural network has a number of $a$ inputs and a number of $b$ outputs. To form models, data records for the $a$ input parameters and the $b$ output parameters are acquired by measuring means.

[0016] If, for example, a model is to be formed for a manufacturing process, the input parameters can be data relating to the materials used, their composition and/or parameters of the production system, for example pressures, temperatures and the like. The resulting product properties, for example, are then measured for the output parameters. In this way, training data records which each contain a set of input parameters and associated output parameters are obtained. The neural network is trained using these training data records, that is to say the weightings of the neurons are adapted iteratively.

[0017] According to one preferred embodiment of the invention, the following definition of the convex envelope is applied as a way of defining the permitted working range:

[0018] $P$ is assumed to be a given, finite set of $n$ points $p_1,...,p_n$. The points $p_i$ (for $i$=1,...,$n$) of the set $P$ are formed by means of the training input data records with which the neural network has been trained. A point $x$, that is to say a specific input data record, is associated with the convex envelope which is formed by $P$ and is referred to as conv(P) if it yields real numbers $\lambda_1,...,\lambda_n \geq 0$ where $\lambda_1 +...+\lambda_n = 1$ so that $\lambda_1 p_1 +...+ \lambda_n p_n = x$ for $p_i \in P$ (for i=1,...,n). (On the theory of convex envelopes, see also: Dieter Jungnickel: "Optimierungsmethoden" [translated title: "Optimization methods"]; Springer, Heidelberg; 1999; ISBN: 3540660577).

[0019] According to one preferred embodiment of the invention, the direct surroundings of the convex envelope are also considered as a permitted working range as neural networks can also supply appropriate results in the direct vicinity of the convex envelope. However, the working range is alternatively restricted directly to the convex envelope as it is not possible to draw a precise conclusion as to where the "direct vicinity" ends. In particular for critical applications which relate, for example, to continuous production, the working range is therefore restricted to the interior of the convex envelope, the external surroundings in the direct vicinity of the convex envelope being excluded from the working range.

[0020] In the practical application, in particular in time-critical applications, it is of particular significance to use efficient procedures in order to determine whether an input data record is in the permitted working range of the associated neural network.

[0021] The algorithms *Quickhull* (see C.B. Barber, D.P. Dobkin and H.T. Huhdanpaa: "The Quickhull Algorithm for Convex Hulls"; ACM Transaction Mathematical Software; Vol. 22, No. 4; 1996; p 469-483, Simplex Algorithm) as well as the simplex algorithm (see, for example, Dieter Jungnickel: "Optimierungsmethoden" [translated title: "Optimization methods"]; Springer, Heidelberg; 1999; ISBN: 3540660577) are known *per se* from the literature. These methods are inefficient in highly dimensional spaces (i.e. input dimensions greater than 9) because they require extremely long computing times and fail on commercially available computers due to the memory requirement. On the other hand, in preferred embodiments of the invention it is possible to have recourse to three basically different, very efficient methods.

[0022] According to one preferred embodiment of the invention, firstly a simplex composed of a number of $d + 1$ non-collinear points from the set $P$ is formed in order to check whether an input data record is in the convex envelope, $d$ being the dimension of the space formed by $P$. A point is then selected from the interior of this simplex. To do this, it is possible to use, for example, the center of gravity of the simplex which is calculated from the vertices of the simplex. This point is referred to below by $x_0$.

[0023] In the next step, the path $[x, x_0]$ between the point $x$ defined by the input data record and the point $x_0$ selected from the simplex is considered. It is then checked whether there is an intersection point of the path $[x, x_0]$ with a facet of the simplex. The facets are the "side faces" of the simplex.

[0024] If there is no such intersection point, this means that the point $x$ is in the interior of the convex envelope.

[0025] If the opposite is the case, this results in the point $x$ being outside the simplex. However, this does not yet answer the question as to whether the point $x$ is inside or outside the convex envelope. It is therefore checked whether it is possible to form a further simplex from $d + 1$ non-collinear points from the set $P$ in such a way that the further simplex contains the intersection point with the facet and a section of the path $[x, x_0]$.

[0026] If this is not possible, the result of this is that the point $x$ is outside the convex envelope (*Caratheodory's* set). If such a simplex can be formed, the check is carried out again to determine whether an intersection point of the path $[x, x_0]$ exists with a facet of the further simplex. As there is only a finite number of points in P after a finite number of iterations, this method indicates whether or not $x$ is in the convex envelope as all the simplices can be checked successively.

[0027] According to one preferred embodiment of the invention, the check to determine whether it is possible to form a further simplex which contains a section of the path $[x, x_0]$ is carried out as follows: firstly, the vertices of the facet which is intersected by the path $[x, x_0]$ are determined. Then a further point is selected from the set $P$. This can be any desired point which is not associated with the vertices of the facet.

[0028] A further simplex is then formed on a trial basis from the further point and the vertices. If this further simplex which is formed on a trial basis contains a section of the path $[x, x_0]$, this further simplex which is formed on a trial basis is used as the simplex for a further iteration of the method.

[0029] If such a section of the path $[x, x_0]$ is not contained in the simplex formed on a trial basis, the further point which is selected from $P$ is replaced by another point in order to form a further simplex on a trial basis, and in order to carry out again the subsequent check to determine whether a section of path $[x, x_0]$ is in the simplex formed on a trial basis.

[0030] This method is carried out until either a further simplex has been found or all the points which are possible from the set $P$ have been selected without a simplex which fulfils the secondary condition of containing a section of the path $[x, x_0]$ having been formed. In this case, the method ends with the conclusion that it is not possible to form a further simplex which contains a section of the path $[x, x_0]$, that is to say $x$ is outside the convex envelope.

[0031] According to a further preferred embodiment of the invention, a different geometric property of the convex envelope is used. This property is as follows:

[0032] If there is a hyper-plane through the point $x$ to be investigated so that all the $p_i \in P$ are located on one side of the plane, the point $x$ is outside the convex envelope formed by $P$ (set of *Hahn-Banach*). If there is no such plane, the point lies in the interior.

[0033] According to a further preferred embodiment of the invention, in order to answer the question as to whether or not a point $x$ is in the convex envelope, it is checked whether the equation system given by the analytical definition of the convex envelope can be solved. For this purpose, an iterative method is used.

[0034] According to a further preferred embodiment of the invention, a model for checking whether an input data record is in a permitted working range of the neural network is positioned before a neural network. If the respective system is a system with a plurality of neural networks and/or a system with rigorous model components, that is to say a so-called hybrid model, such a module is preferably positioned before each neural network of the system. If a plurality of neural networks is used, these modules can be logically linked to a logic “AND” in order to ensure that an input data record is in the permitted working range of all these neural networks. This is significant in particular in hybrid models.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0035] In what follows, preferred embodiments of the invention are explained in more detail with reference to the drawings, in which:

Figure 1 shows a flowchart of a first embodiment of a method for checking whether an input data record is in the convex envelope,

Figure 2 shows a development of the method from Figure 1 for determining a further simplex,

Figure 3 shows a further embodiment of a method according to the invention for checking whether an input data record is in the convex envelope,

Figure 4 shows a graphic illustration of the method in Figure 3,

Figure 5 shows a further embodiment of the method for checking whether an input data record is in the convex envelope, based on a check as to whether there is a solution for the equation system provided by the analytical definition of the convex envelope,

Figure 6 shows a block diagram of an embodiment of a system according to the invention.

## DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

[0036] Figure 1 illustrates a first embodiment of the method for checking whether an input data record is in the convex envelope. This method starts from a point $x_0$ in the interior of the convex envelope and checks whether the path $[x, x_0]$ is in the interior of the convex envelope.

[0037] Here, $x$ is the point which is determined by the input record and it is desired to determine whether this point also lies within the interior of the convex envelope.

[0038] For this purpose it is tested whether the path $[x, x_0]$ intersects one of the facets of the convex envelope. If this is the case, the point $x$ lies outside. Here, use is made of the geometric property of the convex envelope that any linear connection between any two points on the convex envelope lies completely in the convex envelope.

[0039] This method is based on the procedure described below:

A *d*-dimensional space $\mathbf{R}^d$ is assumed, *d* being the number of non-collinear training input data records of the neural network. The set *P* of points includes all the training input data records with which the neural network has been trained. This set P of points is therefore contained completely in the space $\mathbf{R}^d$. In addition, a point $x_0$ will be assumed from the interior of the convex envelope which is formed by *P*, with a known representation as a convex linear combination of the points from P, i.e. there are $\lambda_1^{(0)},...,\lambda_n^{(0)} \geq 0$ where $\lambda_1^{(0)} + ... + \lambda_n^{(0)} = 1$ and $\lambda_1^{(0)} p_1 + ... + \lambda_n^{(0)} p_n = x_0$. According to the set of *Caratheodory*, the coefficients $\lambda_i$ ($i$=1,...,n) can be selected such that all are equal to 0 with the exception of $d$+1. In addition, it is assumed $x \in \mathbf{R}^d$ is a point for which it is to be investigated whether or not it lies in the interior of conv(P).

[0040] It is assumed that $[x, x_0]$ is the path between the points x and $x_0$. The known coefficients $\lambda_i^{(0)}$ ($i$=1,...,n) are then modified in such a way that a linear combination with the new coefficients yields a point $x_1$ which is located on the path $[x_0 x]$. This procedure is repeated until finally the point x is found, or one of the lateral boundaries of the convex envelope is reached.

[0041] In order to modify the coefficients $\lambda_i^{(0)}$, a suitable solution of the following, underdetermined, linear equation system is searched for:

Equation 1

$$\begin{aligned} \sum_{i=1}^{n} \varepsilon_i p_i &= x - x_0 \\ \sum_{i=1}^{n} \varepsilon_i &= 0 \end{aligned}$$

[0042] Then, a factor $c>0$ is determined such that $\lambda_i^{(0)} + c\varepsilon_i \geq 0$ for $i$ = 1,..., n applies. It will then be assumed that $\lambda_i^{(1)} := \lambda_i^{(0)} + c\varepsilon_i$. Then, $x_1 := \lambda_1^{(i)} p_1 + ... + \lambda_n^{(i)} p_n$ is a convex linear combination for a point $x_1 \in$ *conv(P)* which is nearer to $x$ than $x_0$. If the above-described linear combination in which at maximum $d$+1 coefficients are unequal to 0 has been assumed for $x_0$, and if the largest possible c is selected, in this way the intersection point of the path $[x_0,x]$ with a facet of the simplex which is formed by the points from $P$ which are associated with the coefficients is obtained.

[0043] However, the equation system of the equation 1 cannot be solved in a uniquely defined way and a $c > 0$ that fulfils the abovementioned requirements in order to determine new coefficients $\lambda_i^{(1)}$ cannot be found for each solution.

[0044] Furthermore, an iterative method is specified which makes it possible to answer the question as to whether a solution of the equation system exists, and thus whether or not the point $x$ is in the convex envelope:

[0045] Initialization step: it is assumed that $d$ is the dimension of the space in which the convex envelope is located. In order to determine a starting value $x_0$, $d$ random, linearly independent points $q_j^{(0)} \in P$ for $j = 1,\ldots,d$ are selected. Then, a

$$x_0 = \lambda_1^{(0)} q_1^{(0)} + \ldots + \lambda_d^{(0)} q_d^{(0)} \in \mathrm{conv}\left(q_1^{(0)},\ldots,q_d^{(0)}\right)$$

will be selected as the starting value. Furthermore, we assume $i = 0$.

[0046] Iteration step: a ($d$-1)-dimensional hyper-area is uniquely defined in the $\mathbf{R}^{\mathrm{d}}$ through the points $q_1^{(i)},\ldots,q_d^{(i)}$. This hyper-area will be expanded by adding a further point from the set $P$ to form a $d$-dimensional simplex. It will now be assumed that $q_{d+1}^{(i)} \in P$ is the point with the property that the longest possible part of the path $x_i x$ is in the interior of the simplex $q_1^{(i)},\ldots,q_{d+1}^{(i)}$.

[0047] In order to discover this point, it is necessary to solve virtually the same equation system repeatedly, which can be carried out efficiently. If it is not possible to find a further vertex of the simplex, the point $x$ is outside the convex envelope and the method is aborted. Otherwise, the equation system

$$\begin{aligned} \sum_{j=1}^{d+1} \varepsilon_j q_j^{(i)} &= x - x_i \\ \sum_{j=1}^{d+1} \varepsilon_i &= 0 \end{aligned}$$

has a uniquely defined solution ($\varepsilon_1,\ldots,\varepsilon_{d+1}$) and it is possible to select a $c > 0$ with the properties described above. It is then assumed that:

$$\lambda_j^{(i+1)} := \lambda_j^{(i)} + c\varepsilon_j$$

$$x_{i+1} := \sum_{j=1}^{d+1} \lambda_j^{(i+1)} q_j^{(i)}$$

for $j = 1,\ldots, d+1$. It is possible to select $c$ in such a way that either one of the $\lambda_j^{(i+1)}$ (for $j = 1,\ldots,$ d+1) is equal to 0 or $c = 1$ applies. If the case $c = 1$ occurs, the point $x$ is in the interior of the convex envelope and the method can be terminated.

[0048] Otherwise, a further iteration step has to be carried out. As points $q_1^{(i+1)},\ldots,q_d^{(i+1)}$ the $d$ points will be selected from the set $\left\{q_1^{(i)},\ldots,q_{d+1}^{(i)}\right\}$ in which the respectively associated $\lambda_j^{(i+1)}$ ($j$=1,..., $d$+1) is unequal to 0. Then, $i$ is increased by 1.

[0049] If the point $x$ is in the interior of the convex envelope of $P$, the algorithm supplies a convex linear combination to represent the point. If the point lies outside, $d$ points through which a hyper-plane $E$ is determined, which separates the point set $P$ from the point $x$, are obtained. This means that all the points of the $\mathbf{R}^d$ which lie on the same side of $E$ as the point $x$ cannot be associated with the convex envelope. This can be utilized for a multiple evaluation in order to speed up the entire evaluation considerably.

[0050] One form of implementation of this method is illustrated in Figure 1:

[0051] In step 100, an input data record for which a prediction is to be created is input. This input data record for the neural network determines a point $x$.

[0052] In step 101, a number of $d + 1$ non-collinear points is selected from the set $P$.

[0053] In step 102, the index $l$ is set to zero. In step 103, a simplex $S_l$ is formed from the points selected in step 101.

[0054] In step 104, a point $x_l$ is selected from the interior of the simplex $S_l$. The center of gravity is calculated, for example, from the vertices of the simplex $S_l$ in order to obtain the point $x_l$.

[0055] In step 105, a path $[x_l x]$ is defined between $x$ and $x_l$.

[0056] In step 106 it is checked whether an intersection point $x_{l+1}$ of the path $[x_l x]$ with a facet of the simplex $S_l$ is located between $x$ and $x_l$. It is therefore checked whether, starting from $x_l$ on the straight line in the direction of $x$, firstly $x$ or a facet of the simplex $S_l$ is reached.

[0057] If there is such an intersection point $x_{l+1}$ of the path $[x_l x]$ with a facet of $S_l$, this means that the point $x$ is not within the simplex $S_l$. If the opposite is the case, in step 107 there is an output indicating that $x$ is in the convex envelope, as of course it has been determined that $x$ is in the simplex $S_l$, and this is in turn completely inside the convex envelope.

[0058] If, on the other hand, the point $x$ is outside the simplex $S_l$, in step 108 it is checked whether it is possible to find a further simplex $S_{l+1}$ in $P$ which includes both the intersection point $x_{l+1}$ and a section of the straight line $g$. If this is not possible, in step 109 there is an output indicating that $x$ is outside the convex envelope.

[0059] In the opposite case, the index $l$ is increased by one in step 110, and step 106 is carried out again with respect to the further simplex.

[0060] Figure 2 shows a development of the method in Figure 1 for carrying out the check in step 108. In order to carry out this check, the vertices of the facet of $S_l$, on which the intersection point $x_{l+1}$ is located, are firstly determined in step 200.

[0061] In step 201, a further point is selected from $P$ which is not already a vertex of the facet of $S_l$, and which is not collinear with respect to the vertices of the facet.

[0062] In step 202, a simplex $S'$ is formed from the vertices and the further point from $P$.

[0063] In step 203 it is checked whether the simplex $S'$ includes a section of the path $[x_l x]$. If this is the case, the further simplex $S_{l+1}$ which is being searched for is made equal to the simplex $S'$ in step 204. This then also answers the question that it is actually possible to form such a simplex $S_{l+1}$.

[0064] If the check in step 203 reveals that the simplex $S'$ does not contain a section of the straight line $g$, in step 205 it is checked whether all the possible points from $P$ have already previously been selected in step 201. If this is not the case, in step 201 a further point from $P$ which has not yet been previously selected is selected in order to carry out a further iteration of the method.

[0065] If it was not possible to find a simplex $S_{l+1}$ after "trying out" all the points which are possible from $P$, a corresponding item of information is output in step 206. This means at the same time, that the point $x$ is outside the convex envelope.

[0066] In this embodiment it is of particular advantage that in all cases, after a finite number of steps, the method indicates whether or not the input data record is in the convex envelope, and thus in the working range.

[0067] Figure 3 shows a further embodiment of a method for checking whether an input data record is in the convex envelope. This method is not obtained directly from the definition of the convex envelope as a linear combination of the support points. Instead, a different geometric property of the convex envelope is used here, and is also illustrated graphically in Figure 4:

[0068] If there is a hyper-plane through the point $x$ to be investigated so that all the $p_i \in P$ are on one side of the plane, the point $x$ then lies outside the convex envelope formed by $P$. If there is no such plane, the point is in the interior.

[0069] If the plane is represented by means of the normal vector $k$, the condition "all points $p_i \in P$ lie on one side of the plane" can be expressed as follows:

$$k \cdot \ r_i > 0, \ i = 1 \ldots n$$

$r_i = p_i - x$ being the position vectors of the data points in a coordinate system which has the data point to be investigated at the origin.

[0070] Without restricting the generality, inequality can be interrogated with respect to "greater" as the normal vector $-k$ represents the same hyper-plane as $k$. Points on the facets of the convex envelope lead to a scalar product equal to 0 and are thus a component of the convex envelope.

[0071] An optimization method is preferably used for searching for a hyper-plane.

[0072] Here, the following target function is minimized when the normal vector $k$ varies:

$$F = -\min\left(\frac{k \cdot r_i}{|k|}\right)$$

[0073] If the optimum of $F$ is smaller than 0, the point to be investigated lies outside the convex envelope. For points within the convex envelope it is not possible to find a hyper-plane for which $F<0$ applies.

[0074] For the use as an optimization method, various methods are possible, for example the MATLAB routine fminsearch as well as gradient methods, *Levenberg-Marquard* algorithm or an evolution strategy which can also be used in combination with local methods.

[0075] An advantage which is significant for the running time behaviour of the algorithm is that, if a corresponding hyper-plane has been found for a data point, said hyper-plane also constitutes a solution for all the points on the side of the plane lying opposite the convex envelope. If the investigation for membership of the convex envelope is to be carried out simultaneously for a plurality of data points, the method can thus be considerably speeded up.

[0076] Figure 3 illustrates this method by reference to a flowchart. In step 300, the input data record, that is to say the point $x$, is input.

[0077] In step 301, it is checked by means of one or more of the aforesaid methods, whether there is a hyper-plane which contains $x$ and for which $k \cdot \ r_i > 0$, $i = 1,\ldots,n$ applies, where $k$ is the normal vector of the searched-for hyper-plane, and $r_i$ is the difference vector between a point $p_i$ and $x$ provided by a training input data record.

[0078] If there is such a hyper-plane, it follows in step 302 that $x$ is in the convex envelope. In the opposite case, in step 303 information is output according to which $x$ is outside the convex envelope.

[0079] The check in step 301 to determine whether there is a suitable hyper-plane is illustrated in Figure 4. The points $p_i$ located in the grey-hatched area of Figure 4 form a convex envelope 400. The point $x$ is located outside the convex envelope 400. Between the point $x$ and the points $p_i$ there are the difference vectors $r_i = p_i - x$.

[0080] A hyper-plane 401, which is described by the normal vector $k$, runs through $x$. As all the points $p_i$ of the convex envelope 400 are located on the same side of the hyper-plane 400, it follows from this that $x$ is actually outside the convex envelope 400.

[0081] Figure 5 illustrates a further method for checking whether an input data record $x$ is in the convex envelope.

[0082] In this method it is checked whether there is a solution for the equation system which is obtained from the analytical definition of the convex envelope.

Equation 2

$$\lambda_1 p_1 + \ldots + \lambda_n p_n = \widetilde{x}$$

$$\lambda_1 + \ldots + \lambda_n = 1$$

[0083] Here, a solution is searched for so that the secondary conditions $\lambda_i \geq 0$ are fulfilled. In the following method, successive attempts are made to achieve this.

[0084] As in the method in Figures 1 and 2, in this case also an initial solution for $\lambda^{(0)} := \left(\lambda_1^{(0)},\ldots,\lambda_n^{(0)}\right)$ is assumed for which in general the inequality secondary conditions are not fulfilled.

[0085] Equation 2 is written below in matrix form. Then,

Equation 3

$$P^{(0)}\lambda = x$$

is obtained where a line of ones has been added to the vector $x$ and to the dot matrix $P^{(0)}$ respectively.

[0086] Initialization step

We assume $i$=0 and select a random n-dimensional vector

$\lambda^{(0)} = (\lambda_1,\ldots,\lambda_n)$ where $\lambda_1 + \ldots + \lambda_n = 1$

and $\lambda_i \geq 0$.

[0087] Iteration step

[0088] Firstly, we transform the equation 3 by multiplying it on both sides by a matrix $M$. The matrix $M$ is to be selected here in such a way that the lines of the matrix $\hat{P}^{(i)} := M \cdot P^{(i)}$ are ortho-normed (if such a matrix $M$ does not exist, dependent lines in the matrix $P^{(i)}$ can be omitted). In addition, it is assumed that $\hat{x} := M \cdot x$. It is not attempted now to solve the equation system $\hat{P}^{(i)}\lambda = \hat{x}$ directly but instead we start from the known coefficient vector $\lambda^{(i)}$ and assume $\hat{x}^{(i)} := \hat{P}^{(i)}\lambda^{(i)}$. We then search for a solution of the equivalent equation system

$$\hat{P}^{(i)} \cdot \left(\lambda - \lambda^{(i)}\right) = \hat{x} - \hat{x}^{(i)}.$$

[0089] As in most cases this equation system is underdetermined, we search for the solution $\lambda$ so that $\left\|\lambda - \lambda^{(i)}\right\|$ is minimal (where $\|\cdot\|$ designates the *Euclidean* norm). Here, we can make use of the fact that the matrix $\hat{P}^{(i)}$ is ortho-normed. The following applies

$$\lambda = \lambda^{(i)} + \hat{P}^{(i)T} \cdot \left(\hat{x} - \hat{x}^{(i)}\right),$$

[0090] $\hat{P}^{(i)T}$ being the transponent of the matrix $\hat{P}^{(i)}$. If all the components of the coefficient vector $\lambda$ which is found in this way fulfil the secondary conditions $\lambda_i \geq 0$, a convex linear combination for the point $x$ has been found, and the point $x$ is therefore in the interior of the convex envelope. Otherwise, we set all the coefficients which infringe the secondary condition for the rest of the method to zero, and attempt to correct the components which do not infringe the secondary condition in such a way that this step is compensated for. In practical terms, this is brought about by the fact that all the components which infringe the secondary condition, from the vector$\lambda$, and all the associated columns are eliminated from the matrix $\hat{P}$.

[0091] The vector which is obtained in this way with a relatively small dimension and the matrix which is obtained in this way with fewer columns are designated by $\lambda^{(i+1)}$ and $P^{(i+1)}$.

[0092] For the correction, the (smaller) equation system

$$P^{(i+1)}\lambda = \hat{x}$$

must be solved. To do this, a further iteration step is then carried out, $i$ being increased by one, this time with $\lambda^{(i+1)}$ as the starting value. If the equation system cannot be solved, there is no convex linear combination for the point $x$ and the point is located outside the convex envelope.

[0093] As at least one column is always eliminated at each iteration step, the method comes to a result after a maximum of $n$ steps.

[0094] One embodiment of this method is illustrated in Figure 5.

[0095] In step 500, the index $i$ is set to be equal to zero. In step 501, a starting value for the $n$-dimensional vector $\lambda^{(0)}$ which fulfils the secondary conditions is selected. For this purpose, it is possible to select, for example, $\lambda_i = 1/n$.

[0096] In step 502, the matrix $M$ is calculated. On the basis of this, in step 503 the matrix $\hat{P}^{(i)}$ and the vectors $\hat{x}$ and $\hat{x}^{(i)}$ are calculated.

[0097] On the basis of this, in step 504, $\lambda = \lambda^{(i)} + \hat{P}^{(i)^T} \cdot \left(\hat{x} - \hat{x}^{(i)}\right)$ is calculated.

[0098] In step 505, it is checked whether all $\lambda_i(j=1,...,n)$ of the vector calculated in step 504 are $\lambda \geq 0$. If this is the case, in step 506, it follows that the point provided by the input data record is within the convex envelope.

[0099] If the opposite is the case, in step 507 all the components of the vector $\lambda$ and the corresponding columns matrix $P^{(i)}$ which infringe the secondary condition are deleted. This results in the smaller equation system $P^{(i+1)}\lambda = \hat{x}$.

[0100] In step 508, the index *i* is incremented in order to carry out a further iteration of the method.

[0101] Figure 6 shows a block diagram of an embodiment of a system 600 according to the invention. The system 600 has an input module 601 for inputting an input data record which is composed of a=3 parameters in the example considered here.

[0102] The input module 601 is logically linked to a module 602 which is used for checking whether an input data record lies within the convex envelope of the neural network 603. This checking is carried out, for example, according to a method which is described with respect to Figures 1 to 5, or according to another method.

[0103] The module 602 is logically linked to the neural network 603. If the module 602 determines that an input data record is in the permitted working range of the neural network which is provided by the convex envelope, this input data record is input into the neural network 603, which then outputs at least one predicted value at its output 604. On the other hand, if the module 602 determines that the input data record is not in the permitted working range, a corresponding signal is emitted at the output 605, after which no reliable prediction is possible for the current input data record.

[0104] In addition to the neural network 603, the system 600 can also contain further neural networks (hybrid model), each of which having in turn arranged upstream of it a module which corresponds to the module 602. The results of the individual modules 602 must then be logically linked to a logic "AND". This ensures that all the neural networks of the hybrid model 600 are operated in a permitted working range for a specific input data record of the input module 601. In addition, the system 600 can also contain rigorous model components.